Lab Report of Python Programming

Lab 3: Selections, Loops, Strings and Regular Expressions  
Credit hour: 2

Student Name: 徐政辉

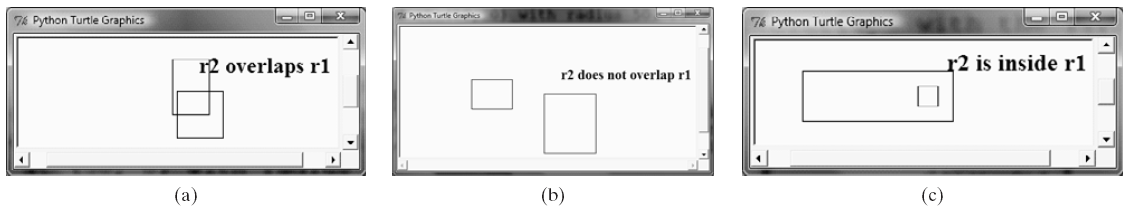
Student ID:2017329621139

## ￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣￣

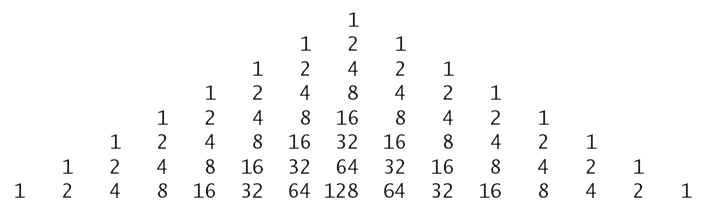
1. Objective
   1. Know how to write a program using if-else;
   2. Know how to write a program using while and for loops;
   3. Know how to create strings and use functions, operators and methods of strings;
   4. Know the syntax and application of the regular expressions.
2. Lab content
   1. (Game: pick a card ) Write a program that simulates picking a card from a deck of 52 cards. Your program should display the rank (Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King) and suit (Clubs, Diamonds, Hearts, Spades) of the card. Here is a sample run of the program:

The card you picked is the Jack of Hearts

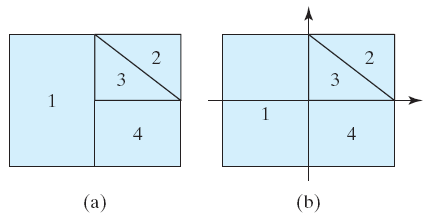
* 1. (Geometry: two rectangles) Write a program that prompts the user to enter the center x-, y-coordinates, width, and height of two rectangles and determines whether the second rectangle is inside the first or overlaps with the first, as shown below



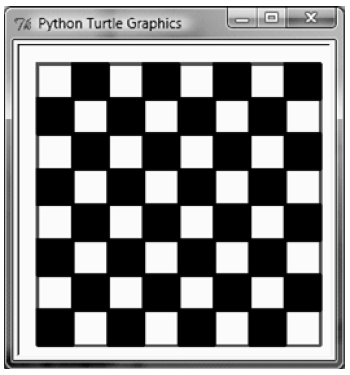
* 1. (Display numbers in a pyramid pattern) Write a nested for loop that displays the following output:



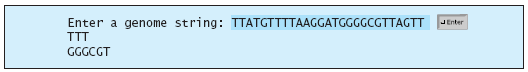
* 1. (Monte Carlo simulation) A square is divided into four smaller regions as shown in (a). If you throw a dart into the square one million times, what is the probability for the dart to fall into an odd-numbered region? Write a program to simulate the process and display the result.  
     Hint: Place the center of the square in the center of a coordinate system, as shown in (b). Randomly generate a point in the square and count the number of times for a point to fall in an odd-numbered region.



* 1. (Turtle: chessboard) Write a program to draw a chessboard.



* 1. (Bioinformatics: find genes) Biologists use a sequence of letters A, C, T, and G to model a genome. A gene is a substring of a genome that starts after a triplet ATG and ends before a triplet TAG, TAA, or TGA. Furthermore, the length of a gene string is a multiple of 3 and the gene does not contain any of the triplets ATG, TAG, TAA, and TGA. Write a program that prompts the user to enter a genome and displays all genes in the genome. If no gene is found in the input sequence, the program displays no gene is found. Here are the sample runs:





* 1. By using regular expressions to extract the phone number from the string.
  2. Assume there are some English texts and some word appears in this text twice. Write a program to check the duplicate word and keep one. For example, the English text is: “This is is a desk.” and the output of the program should be “This is a desk.”

1. Code list

2.1.py

rank=["Ace","2","3","4","5","6","7","8","9","10","Jack","Queen","King"]

suit=["Clubs","Diamonds","Hearts","Spades"]

print("The card you picked is the %s of %s."%(suit[0],rank[0]))

2.2.py

import turtle

x1=eval(input("enter the x coordinate:"))

y1=eval(input("enter the y coordinate:"))

width1=eval(input("enter the width of the rectangle:"))

height1=eval(input("enter the height of the rectangle:"))

x2=eval(input("enter the x coordinate:"))

y2=eval(input("enter the y coordinate:"))

width2=eval(input("enter the width of the rectangle:"))

height2=eval(input("enter the height of the rectangle:"))

def drawrec(x,y,width,height):

turtle.penup()

turtle.goto((x,y))

turtle.pendown()

turtle.goto((x+width,y))

turtle.goto((x+width,y+height))

turtle.goto((x,y+height))

turtle.goto((x,y))

turtle.penup()

drawrec(x1,y1,width1,height1)

drawrec(x2,y2,width2,height2)

turtle.goto((x1+width1,y1))

turtle.pendown()

if x1<=x2 and y1<=y2 and width1+x1>width2+x2 and height1+y1>height2+y2:

turtle.write("r2 is inside r1")

elif width1/2+width2/2<abs(x1-x2) and height1/2+height2/2<abs(y1-y2):

turtle.write("r2 is not overlap r1")

else:

turtle.write("r2 is overlap r1")

turtle.done()

2.4.py

for i in range(8):

print(" "\*(7-i)\*4,end="")

for k in range(i):

print("%4d"%2\*\*k,end="")

for k in range(i+1):

print("%4d"%2\*\*(i-k),end="")

print()

2.5.py

import turtle

turtle.pensize(2)

turtle.color("black")

turtle.penup()

turtle.goto((-240,-240))

turtle.pendown()

turtle.goto((240,-240))

turtle.goto((240,240))

turtle.goto((-240,240))

turtle.goto((-240,-240))

def draw\_sq(x,y,z):

turtle.penup()

turtle.goto((x,y))

turtle.begin\_fill()

turtle.pendown()

turtle.goto((x+z,y))

turtle.goto((x+z,y+z))

turtle.goto((x,y+z))

turtle.goto((x,y))

turtle.end\_fill()

x=-240

y=-240

while y<240:

while x<240:

draw\_sq(x,y,60)

x+=120

y+=120

x=-240

y=-240

while y<240:

while x<240:

draw\_sq(x+60,y+60,60)

x+=120

y+=120

x=-240

turtle.done()

2.6.py

import re

str=input("Enter a genome string: ")

m=re.compile("ATG(.\*?)(TAG|TAA|TGA)")

for x in m.findall(str):

print(x[0])

2.7.py

import re

str=input()

m=re.compile("[1-9][0-9]{10}")

for x in m.findall(str):

print(x)

2.8.py

str=input().split()

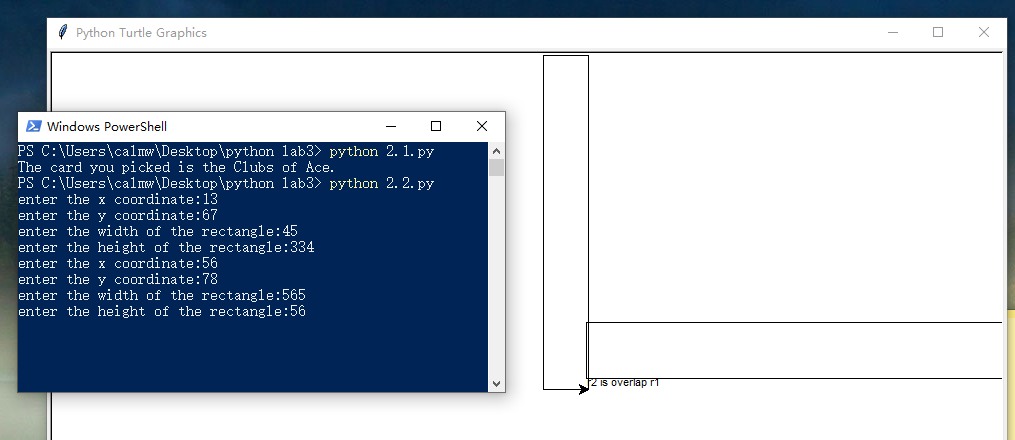
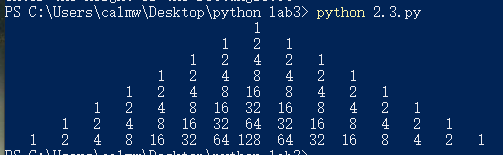
for x in range(len(str)-1):

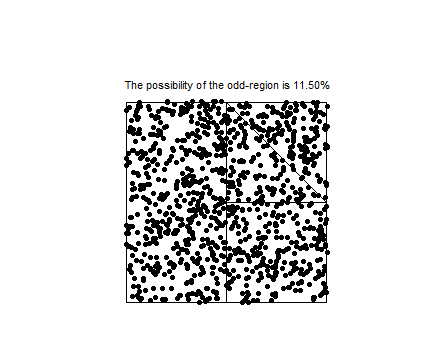
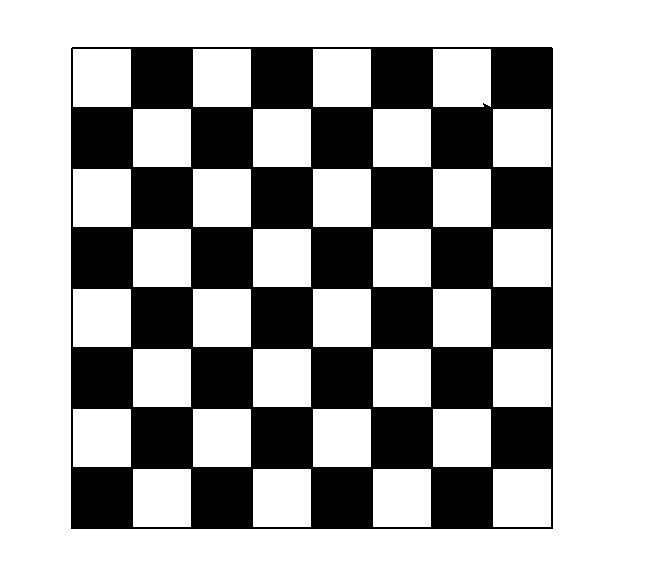
if str[x]!=str[x+1]:

print(str[x],end=" ")

print(str[len(str)-1])

1. Output







1. Analysis and conclusions

After this lab , I have reviewed the regular expressions and have a deep learning about turtle and the base use of loops. There is still many things to do after the lab. I know the python is easy to be used as a web spider. It can catch amount of data from the web sites, so I will learn how to do so in the next study .